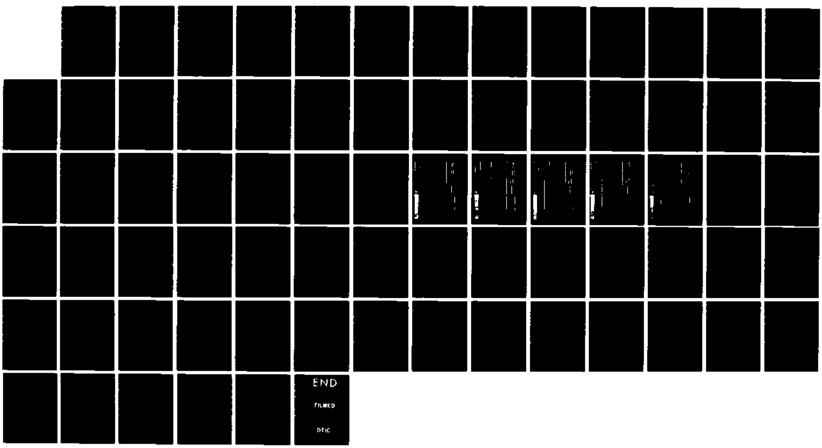
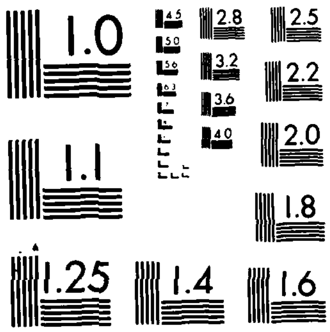
AD-A159 126 A LISP MACHINE FACILITY FOR EXPERT SYSTEMS RESEARCH(U) 1/1
OHIO STATE UNIV RESEARCH FOUNDATION COLUMBUS
B CHANDRASEKARAN AUG 84 AFOSR-TR-85-0636 AFOSR-83-0300

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

RF Project 763725/715560
Final Report

A LISP MACHINE FACILITY FOR EXPERT SYSTEMS RESEARCH

B. Chandrasekaran
Department of Computer and Information Science

For the Period
August 1, 1983 - July 31, 1984

U.S. AIR FORCE
Air Force Office of Scientific Research
Bolling Air Force Base, D.C. 20332

Grant No. AFOSR-83-0300

August 1984

OSU

The Ohio State University
Research Foundation
1314 Kinnear Road
Columbus, Ohio 43212

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A157 120

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR-85-0636
6a. NAME OF PERFORMING ORGANIZATION Ohio State University Research Foundation / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code) 1314 Kinnear Road Columbus, OH 43212	7b. ADDRESS (City, State and ZIP Code) Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR / 8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-83-0300
8c. ADDRESS (City, State and ZIP Code) Bolling AFB DC 20332-6448	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: 61102F; PROJECT NO.: 2917; TASK NO.: A5; WORK UNIT NO.:

11. TITLE (Include Security Classification)
A Lisp Machine Facility for Expert Systems Research

12. PERSONAL AUTHOR(S)
B. Chandrasekaran

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	FROM 1 Aug 83 TO 31 Jul 84	August 1984	72

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Expert systems, types of problem solving, expert systems for design
XXXXXXXXXXXX			

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

During the year, progress was made in a number of directions: 1. A better understanding of different types of problem solving that underlie expert reasoning was obtained. 2. Advances in representing design knowledge as plans in design specialists were made. 3. CSRL, the language for diagnostic expert system building that was designed in our Laboratory, was applied to the implementation of a diagnostic system for the fuel system of an automible and directions for new constructs for the language were obtained. 4. A representation for functional understanding of how a device works was obtained, and methods of automatically generating diagnostic expert systems from this representation of a device were also obtained. 5. An analysis of how techniques and tasks can be matched in expert design was undertaken.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Robert N. Buchal	(202) 767-4939	NM

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

A Lisp Machine Facility for Expert Systems Research

INTRODUCTION

AFOSR Grant 83-0300 was awarded to The Ohio State University Research Foundation, with Prof. B. Chandrasekaran as the Principal Investigator, under the DOD-University Research Instrumentation Program. The award was for the purchase of four LISP machines and peripherals in order to advance our capability to do artificial intelligence research in general and expert systems (or knowledge-based systems) research in particular. We are also simultaneously in receipt of a research grant from AFOSR, AFOSR 82-0255, for investigations of expert system principles for diagnostic reasoning and applications to databases. Thus the Lisp machine award meshes in nicely with the basic research program that we have been pursuing.

EQUIPMENT ACQUIRED

With cost-sharing by the Ohio State University, the following equipment was acquired:

- 4 Xerox EA 8000 NS, processors w. 1.5 MB memory, 8014 workstations, (42 MB disk), 2-button mouse, programmer's keyboard, RS232 port, Interlisp programming environment, ethernet tranceiver (these workstations are informally called Xerox Dandelion Lisp machines)

- Ethernet connection for the departmental VAX 11/780, which is to be used as the file server

In addition, we have set aside funds to purchase an ethernet coonection for the departmental DEC20/60 on which substantial AI research also is performed. The Lisp machines, the VAX and the DEC20/60 are in varying stages of connection into an ethernet configuration.

REPORT ON RESEARCH ACTIVITIES

As would be expected, quite a bit of the early months was spent familiarizing the research group with the Interlisp environment, and learning the hardware. We were also chosen by Xerox Palo Alto Research Center as a

A-1

Beta test site for a knowledge programming software system called LOOPS, and thus we spent a considerable amount of time learning and testing the software system. The environment is now fully functional, and the research group is completely familiar and at home with the system.

Even though only a limited time has been available to pursue new research activities in the new environment, we are glad to report considerable research progress.. We are now using the machines almost exclusively for our AI research, except for those projects where DEC20 or the VAX affords greater portability.

Three projects that have been done using the Lisp machines are briefly described here, and papers describing the projects are enclosed as appendices.

1. <u>MDX/MYCIN: The MDX Paradigm Applied to the Mycin Domain</u>. Comparison of different approaches to expert system design for a given task, such as diagnosis, is difficult since they are often embodied in systems for domains with very different characteristics. It is a priori difficult to decide if a given difference in the approaches is necessitated by the differences in the domain. For example, it might be suggested that MYCIN's <u>global and numeric uncertainty calculus</u> is needed in domains such as MYCIN's, apparently characterized by a great deal of uncertainty in knowledge and data, while the approach of MDX, another medical system, which uses local combinations of qualitative probabilities only may be too weak in such domains. In order to study the relationship between the domain characteristics and problem solving approaches of the two systems, we constructed an MDX-like system for a subdomain of MYCIN, and conducted a number of experiments on the resulting system. The results demonstrate that the MDX paradigm is effective in this domain, and, additionally, offers knowledge engineering advantages along the dimensions of debugging ease and system extensibility.

   The Lisp machine was used for this research and was especially useful for a number of reasons, including the power of the knowledge environment for construction of the system, and ease of debugging due to the graphics, windowing and mouse capabilities.

   A paper describing this research will shortly appear as an invited paper in <u>International Journal of Computers in Mathematics</u>, special issue on artificial intelligence applications.

2. <u>The CSRL Language for Diagnostic System Construction</u>: We have implemented CSRL (Conceptual Structures Representation Language) in

the Interlisp/Loops environment in the Xerox Lisp machine. This language had earlier been implemented in UCI-Rutgers Lisp on the DEC20/60. This language facilitates the development of expert diagnosis systems based on a paradigm of "cooperating diagnostic specialists." In our approach, diagnostic reasoning is one of several generic tasks, each of which calls for a particular organizational and problem solving structure. A diagnostic structure is composed of a collection of specialists, each of which corresponds to a potential hypothesis about the current case. They are organized as a classification or diagnostic hierarchy, e.g., a classification of diseases. A top-down strategy called establish-refine is used in which either a specialist establishes and then refines itself, or the specialist rejects itself, pruning the hierarchy that it heads. CSRL is a language for representing the specialists of a diagnostic hierarchy and the diagnostic knowledge within them. The diagnostic knowledge is encoded at various levels of abstractions: message procedures, which describe the specialist's behavior in response to messages from other specialists; knowledge groups, which determine how data relate to features of the hypothesis; and rule-like knowledge, which is contained within knowledge groups.

The availability of CSRL in the Lisp machine has already made it possible to construct a number of diagnostic expert systems, and use them as the basis for further research on diagnostic reasoning. We describe one such research in the next paragraph. We enclose as an appendix a paper describing the CSRL system which will appear in International Journal of Computers in Mathematics, special issue on artificial intelligence applications.

3. Assembling the Best Explanation: Going from data describing a situation to an explanatory hypothesis that best accounts for the data is a commonly occurring knowledge-based reasoning problem. Sometimes the need is to assemble interacting hypothesis parts into a unified hypothesis. In a medical diagnosis, for example, there might be several diseases present, and they might be related causally. Disease hypotheses sometimes overlap in what they can explain.

In this research we have developed a general mechanism for accomplishing the unification of sub-hypotheses with possibly overlapping domains of explanation. This mechanism makes use of plausibility information concerning the sub-hypotheses, along with information about what a sub-hypothesis can explain in the particular situation, to build towards a complete explanation. The novel capability arises of confirming a sub-hypothesis on the basis of its ability to explain some feature for which there is no other plausible explanation.

Hypothesis interactions are considered to be of two general types, each with its own kind of significance for the problem-solving:

- Explanatory interactions, i.e. due to overlapping in what they can account for.

- Substantive interactions of mutual support and incompatibility, e.g. resulting from causal, logical, or definitional relations. (Two disease hypotheses might offer to explain the same findings without being especially compatible or incompatible causally, logically, or definitionally. On the other hand, hypotheses might be mutually exclusive, e.g. because they represent distinct sub-types of the same disease. These two senses in which hypotheses may be said to be "alternatives" need to be distinguished, so that the problem solving can be organized appropriately.)

The mechanism we describe can accommodate additive cooperation in accounting for the features of the situation. While this is not yet general enough to handle all types of explanatory interaction, it is nevertheless more general than the set-covering model which considers that an hypothesis either fully accounts for a feature or it does not. The mechanism described here also accommodates substantive hypothesis interactions of mutual compatibility and incompatibility, and interactions of the sort where one hypothesis, if it is accepted, suggests some other hypothesis. Prospects seem good for extending the mechanism to accommodate other forms of interaction too.

This mechanism has been used successfully as the basis for an expert system, RED, designed to solve real-world problems of red-cell antibody identification. We are enclosing a paper which will appear in the December 1984 Proceedings of the IEEE Expert Systems Workshop to be held in Denver, Colorado. The implementation of this system was done using CSRL, Loops and Interlisp in the Lisp machines that we have acquired.

CONCLUDING REMARKS

As can be seen, even though it has been only a year since the grant was awarded, we have ordered, acquired, and installed the Lisp machine facility, and conducted a number of research projects using the machines. We expect that this facility will continue to be used for a number of basic advances in expert systems research, especially in support of the research program that is being supported by a companion research grant from AFOSR.

APPENDIX 1

To appear in the Special Issue of
International Journal of Computers and
Mathematics on "practical artificial
intelligence systems."

TITLE MDX-MYCIN:
The MDX Paradigm Applied
To The Mycin Domain

Authors Jon Sticklen, B. Chandrasekaran
AI Group
Department of Computer
and Information Science
Ohio State University
Columbus, Ohio 43210

J.W. Smith, John Svirbely
Department Of Pathology
School Of Medicine
Ohio State University
Columbus, Ohio 43210

Send Proofs To: Jon Sticklen
AI Group
CIS Department
2036 Neil Avenue Mall
Ohio State University
Columbus, OH 43010
(614) 422-1413

ARPA Address: STICKLEN@RUTGERS

MDX-MYCIN:

The MDX Paradigm Applied To The Mycin Domain

Jon Sticklen, B. Chandrasekaran,
J.W. Smith, John Svirbely

Abstract

Comparison of different approaches to expert system design for a given task, such as diagnosis, is difficult since they are often embodied in systems for domains with very different characteristics. It is a priori difficult to decide if a given difference in the approaches is necessitated by the differences in the domain. For example, it might be suggested that MYCIN's global and numeric uncertainty calculus is needed in domains such as MYCIN's, apparently characterized by a great deal of uncertainty in knowledge and data, while the approach of MDX, another medical system, which uses local combinations of qualitative probabilities only may be too weak in such domains. In order to study the relationship between the domain characteristics and problem solving approaches of the two systems, we constructed an MDX-like system for a subdomain of MYCIN, and conducted a number of experiments on the resulting system. The results demonstrate that the MDX paradigm is effective in this domain, and, additionally, offers knowledge engineering advantages along the dimensions of debugging ease and system extensibility.

MDX-MYCIN:

The MDX Paradigm Applied To The Mycin Domain

Jon Sticklen, B. Chandrasekaran,
J.W. Smith, John Svirbely

1. Introduction

There are a number of approaches to diagnostic expert system design available; typically they are implemented in domains which differ in their characteristics. It is a priori difficult to comparatively evaluate the approaches, since it is often unclear which facets of the differing approaches are functions of the domain differences. While in principle it is sometimes possible to compare the approaches, often the only reliable method seems to be to apply the different techniques to a carefully chosen domain and test the resulting systems. The only study in this vein of which we are aware is the work of Sherman [1].

In this paper, we will compare two methodologies: MDX and MYCIN. This will be done by applying both approaches to the same domain; viz that of infectious meningiococcal disease, a subdomain of MYCIN which comprises roughly 40% of the total MYCIN system. The comparisons will be along two dimensions: formal performance comparisons, and informal comparisons along ease of construction, debugging, and system extension criteria.

2. Comparisons Of Methodologies And Domains

2.1. The Methodologies

The MYCIN [2] approach is too well-known to need a recapitulation here. Use of production rules, a separate inference engine, and a global uncertainty calculus are some of the essential aspects of the approach.

The MDX family of systems [3, 4] comprise an expert system for medical diagnosis in the cholestatic liver domain. MDX itself is the diagnostic component.

2.1.1. The MDX Classification Hierarchy

The MDX paradigm for diagnosis relies on the decomposition of diagnostic knowledge into a classification hierarchy of cooperating specialists. The control regime follows an "establish-refine" mechanism in which each specialist, when invoked, attempts to determine if the signs and symptoms of the current patient are consistent with the diagnostic hypothesis this specialist represents (i.e., establish itself) and then to call on its subspecialists to further classify the current findings (i.e., refine itself). It is "establish-refine" that allows MDX to cut through some of the combinatoric difficulties and prune the diagnostic problem space.

2.1.2. MDX Specialist Decomposition Into Knowledge Groups

All decision knowledge about a given diagnostic hypothesis is contained in the form of "knowledge groups" within the corresponding specialist. Hence, all reasoning that leads to the determination of the confidence in the validity of of a given hypothesis is done locally within the specialist representing that diagnostic category.

These knowledge groups within each specialist are also organized into a hierarchy. It is important to note that the MDX approach involves two distinct hierarchies:

1. the hierarchy of specialists and
2. the hierarchy of knowledge groups within each specialist.

In this section we are dealing with the hierarchy of knowledge groups within each specialist.

To make more concrete the way in which the knowledge groups of an individual diagnostic concept interact to deal with and abstract uncertain case data, consider Figures 1a, 1b, and 1c below, which show three knowledge groups for a diagnostic concept "Battery Problem", that could exist in a automotive diagnostic system.*

PLACE Figure 1a HERE.

PLACE Figure 1b HERE.

PLACE Figure 1c HERE.

The diagnostic concept BatteryProblem has only two levels of knowledge groups: at the top level, a group called batteryProblem.TopLevel (Figure 1a), and at the tip level two

*The form here is for pedagogical purposes; the actual implementation form of knowledge groups in MDX-MYCIN is a modified truth table.

groups called batteryProblem.Observations (Figure 1b) and batteryProblem.History (Figure 1c). When BatteryProblem itself is called upon by its superior concepts to establish itself, it would "invoke" its top level knowledge group, batteryProblem.TopLevel. Then batteryProblem.TopLevel would call upon its subordinate knowledge groups batteryProblem.Observations and batteryProblem.History, the results of which are used in the top level knowledge group.

Now consider the knowledge group batteryProblem.History. This knowledge group reaches a conclusion based on two pieces of "patient-specific data": what is the current season and how old is the battery. These pieces of data would be made available to the knowledge group via calls to an auxiliary database specialist, or by directly asking the user. At invocation, each row of the knowledge group is examined in turn until the conditions of one row are found to be true, in which case the knowledge group returns the conclusion of that row. For example, in batteryProblem.History, if (season = winter) AND (battery age > 5 yrs) then it returns a result of +3. If no row of a knowledge group matches the current case data, then the knowledge group returns a symbolic value of 0, indicating that it is unable to utilize any of its domain knowledge in the current case.

Note that each row of a knowledge group could be viewed as a single production rule. But also note that the result of a knowledge group is a single <u>symbolic</u> distillation of the importance of this knowledge group for the current case. And further, that this one symbolic measure is made available for

[7]; thanks to the LOOPS team, Mark Stefik, Danny Bobrow, and Sanjay Mittal, at Xerox PARC for many helpful suggestions.

Lastly, our thanks to the reviewers for their useful comments.

This work has been supported by NSF grant MCS-8103480, and Biomedical Computing and Information Processing Training Grant NLM 5T15LM07023-05.

The specific technical conclusions that can be drawn from this study are:

- The MDX methodology can offer pruning advantages under appropriate conditions because of its hierarchical structure. However, the strict establish-refine regime requires modification in domains where there may be a dearth of knowledge for the establishment of intermediate nodes.

- Even in domains characterized by high degrees of uncertainty, the MDX method of qualitative, concept-dependent likelihood combination at local levels can be successful.

- MYCIN offers a certain kind of modularity in knowledge acquisition in the form of rules. However, the MDX methodology offers certain other kinds of modularity that the knowledge engineer can directly use.

In a broader perspective, it is important to understand that conclusions on two levels have been reached. The first level is that of computational adequacy: the method of carrying on likelihood calculations on a symbolic, local basis is adequate for the computational task of diagnosis in the MYCIN domain. The second level is that of naturalness of expression: the MDX methodology affords the knowledge engineer a framework for building classification-type diagnostic systems which lead to a number of desirable system traits for both system extension and system debugging.

7. Acknowledgments

Many thanks to Prof. Ted Shortliffe for supplying the rule set of MYCIN, for supplying information on how to run MYCIN at SUMEX, and for insightful comments on an earlier draft.

The implementation language for MDX-MYCIN is the LOOPS language

this domain not all the intermediate nodes contain knowledge that would allow establishing the node in the face of missing lab data. In fact, when lab data is not present, these intermediate nodes can be established only on the basis of the establishment of one of their children nodes.

This of course is not permitted in the original MDX framework. But it can be done in MDX-MYCIN at those nodes which are known to be deficient in establish knowledge. We needed to modify the control strategy to permit passing of control to the children whenever necessary patient specific data was not present. In a sense, this can be considered a "second pass" in problem solving where a more exploratory approach is employed.

6. Discussion

In order to place the comparison in proper focus, it is essential to reiterate that MYCIN was designed as a diagnostic and therapy-recommendation system, while MDX-MYCIN's establish-refine problem solving limits it to the diagnostic task. If the MDX framework were to be applied to the total task of MYCIN, the therapy task would be handled by a separate knowledge structure with a different problem solving strategy and a separate specialist structure.

The methodology adopted here for comparative evaluation, viz, applying the different approaches to the same domain, can be expensive in time if care is not exercised to select a target domain small enough to be practical, yet complex enough to be representative.

established itself at the appropriate level given the case data. The knowledge groups used internally in StaphCoagPos can now be added (as shown in Figure 6), and again we can look at the values returned by the knowledge groups (as shown in Figure 7).

PLACE Figure 6 HERE.

PLACE Figure 7 HERE.

Now the trail can quickly lead to a single knowledge group within one diagnostic specialist. Figure 8 depicts the final step in this process. Assuming that the StaphCoagPos.cancer knowledge group has been singled out by the medical expert for possible revision, here we see the cancer knowledge group put up in an editing window for revision.

PLACE FIGURE 8 HERE.

The process would not, of course, actually terminate now. Still the altered decision knowledge would have to be tested in concert with the medical expert over both this case and others. BUT it can be tested in isolation because the knowledge about cancer that is relevant to StaphCoagPos is abstracted in a single knowledge group.

5. Extensions to MDX Point of View

The design of MDX-MYCIN revealed a limitation in the "establish-refine" strategy for classification. In particular, in considered.

Again, in MYCIN this is not the case. Adding new decision knowledge for an existing disease hypothesis in MYCIN would force an examination of much of the entire rule base.

With respect to ease of debugging, consider a typical "knowledge engineering" session for MDX-MYCIN involving a knowledge engineer who is in the test phase of building an expert system with a collaborating medical expert. The following five figures are direct screen images from a XEROX 1108 running the MDX-MYCIN system which would be typical views for such a knowledge engineering session.

Assume that a test case is selected to run on MDX-MYCIN. In Figure 4,the MDX-MYCIN hierarchy just after the test run is completed is shown near the bottom of the view. The objects shown in reverse video are those diseases in the classification hierarchy which have been established.

PLACE Figure 4 HERE.

Figure 5 shows the addition of establishing values of each of the diagnostic specialists depicted in Figure 4.*

PLACE Figure 5 HERE.

At this point let us assume the resident medical expert points out that (for example) the tip level node StaphCoagPos has not

---

*MDX-MYCIN uses an arbitrary scale of (-3, +3) with +3 being near certainty that the disease is present.

assumes, this addition can be made in a modular fashion.*

This is not true if the MYCIN approach is followed. Adding a new disease hypothesis to MYCIN would require a review of the entire database of existing rules. This is because the single layered rule set of MYCIN does not explicitly incorporate a notion of context in which the individual rules will be examined. The context must be set as clauses in each rule itself, typically by some sort of clause like "If you are currently looking at meningitis AND ..."

A similar argument also is true along the dimension of internal extensibility. To add new decision knowledge for an existing disease hypothesis in MDX-MYCIN, we need first only consider the specialist *representing* that disease. Then we go on to consider the knowledge groups within that specialist. If the new decision knowledge fits into one of the existing groups, it is added there. An important reason the knowledge groups are <u>named</u> objects is to facilitate the identification of the proper knowledge group into which new decision knowledge should fit. If the new decision knowledge does not fit into an existing knowledge group, a new knowledge group is created. In either case, the bulk of the decision knowledge for the disease hypothesis in question is left untouched, and need not be

---

*While there still may be interactions between the newly added module and those in the current structure, the methodology <u>demands</u> that those interactions be explicitly identified and taken into account. This aspect of the MDX methodology involves the use of a blackboard and goes beyond the scope of this paper; a full discussion can be found in Gomez [8].

an efficient pruning of its diagnostic tree so that only a subset of the tip level nodes are examined. The intermediate layers of the diagnostic tree make use of lab data to quickly focus on the part of the tree from which the final diagnosis will come.

MYCIN on the other hand does not contain diagnostic hypotheses that correspond to the intermediate node in MDX-MYCIN, hence the pruning characteristics exhibited in MDX-MYCIN are not present in MYCIN.

4. Knowledge Engineering Issues

In addition to the performance considerations listed in the above section, the dimension of knowledge engineering is relevant. Let us consider the following metrics for judging the utility of expert systems: debugging ease and extensibility. Ease of debugging is self explanatory. Extensibility has two components: the ease with which new decision knowledge is added to the system for one of the existing diagnostic hypotheses (internal extensibility), and the ease with which totally new diagnostic hypotheses are added to the system (external extensibility).

To externally extend MDX-MYCIN, we need only add new diagnostic specialists. Note that a specialist in an MDX classification tree corresponds directly to a diagnostic hypothesis. To add a new diagnostic category we would, of course, have to gather new decision knowledge for the new hypothesis possibility. Because of the hierarchical decomposition that the MDX methodology patient's meningitis. Hence, an appropriate set of metrics to measure both MYCIN and MDX-MYCIN are

- Metric 1: whether or not the literature diagnosis is contained in the output list (scored +1 if contained, 0 if not)
- Metric 2: the number of other entries in the output list

There are two sets of data for each case from the medical literature. One set for the initial presentation of the patient, and a second set that includes data that would be available to the diagnostician at a later time. This reflects the two typical situations in which the human infectious disease expert finds herself. The data that is added later is culture data from the microbiology lab. Both the initial presentation and the full data situations require diagnoses in the setting of potential cases of meningitis because of the severity of the disease; i.e., to be successful, treatment must begin as soon as possible, even without a thoroughly reliable diagnosis. The results are shown in Figure 3.

PLACE Figure 3 HERE.

3.3. Operation of MDX-MYCIN: Pruning Efficiency

When analyzing the parts of the MDX-MYCIN diagnostic tree that are activated in different cases, it became clear that when an initial presentation case is input to the system, almost all of the tip level nodes in the tree are activated. On the other hand, when a case containing full lab reports are input, MDX-MYCIN does

Cases From The MYCIN Library

For the cases from the MYCIN library, the following metrics were used:

- Metric 1: The percentage of diseases in the output list of MYCIN that are also in the output list of MDX-MYCIN. The closer to 100%, the better the fit between the two systems.
- Metric 2: The percentage of diseases in the output list of MDX-MYCIN that are not in the output list of MYCIN. The lower this number, the better the fit between the two system. This second metric really is a check that MDX-MYCIN does not apply a shotgun technique to finding a list of likely diseases.

The results from the MYCIN library cases shows the averages for the two metrics listed above to be the following: average metric 1 = 95%, and average metric 2 = 31%

It should be added that our medical informants carefully analyzed why metric 2 above was larger than expected. The conclusion of their analysis was that MDX-MYCIN would on occasion include in its output list a likely hypothesis not in the list of MYCIN that an infectious disease specialist would not refute on medical grounds. Over the cases of meningitis cases in the MYCIN library, we believe the results for these two metrics demonstrate the applicability of the MDX approach in the MYCIN domain.

Cases From the Medical Literature

For the cases from the medical literature* , we ran the test data both on MYCIN and on MDX-MYCIN. The cases taken from the literature list just one "right" answer for the diagnosis of the

*References to the cases we used may be obtained on request.

groups, the columns could deal with the results from lower level knowledge groups.

PLACE FIGURE 2 HERE.

Note again that knowledge groups within an MDX system abstract patient specific knowledge (or the results of other knowledge groups) into a single symbolic result that can be used by higher level knowledge groups to provide an eventual overall answer for the diagnostic concept itself.

After assembling and structuring all the necessary domain knowledge, MDX-MYCIN was implemented in the LOOPS language [7].

3.2. Formal Evaluation Of Results

To test the MDX-MYCIN system, we used three sets of cases as follows: five cases of meningitis from the MYCIN library, fifteen cases of meningitis from the medical literature with signs and symptoms of the patient at presentation, and the same fifteen cases with additional laboratory data available (specifically organism aerobicity, gram stain, and genus).

To understand the results from MDX-MYCIN note that at the conclusion of the diagnostic phase, MYCIN produces a list of diseases it considers to be likely possibilities. The therapy selection part of MYCIN takes this list and tries to "cover" for all the diseases with the minimum number of antimicrobial agents. In like manner, MDX-MYCIN produces a list of likely diseases. To compare the operation of the two systems, we compare the two output lists of likely diseases.

3. MDX-MYCIN

In this paper we will describe the MDX-MYCIN system, an expert system for diagnosis in the domain of infectious meningitis. MDX-MYCIN is constructed following the MDX paradigm. It is important to note that MYCIN was not constructed to perform diagnosis only, it also undertakes therapy recommendation. In building MDX-MYCIN, we concentrate solely on the diagnostic task; comparisons made between MYCIN and MDX-MYCIN deal only with diagnosis.

3.1. Constructing the System

The domain knowledge necessary for infectious meningitis was selected from the corpus of MYCIN rules which we obtained courtesy of Dr. Ted Shortliffe, the designer of MYCIN. In cooperation with expert medical personnel, we constructed a classification hierarchy for the area of meningiococcal disease. We then distributed the MYCIN rules for meningitis among the diagnostic specialists in the classification hierarchy for MDX-MYCIN. Next the rules falling within each MDX-MYCIN specialist were further factored into meaningful, named knowledge groups, such as the one in Figure 2 for the knowledge group representing lab data that is used to conclude that the organism causing meningitis is e. Coli. The right column in Figure 2 shows a symbolic degree of certainty on a scale from -3 to +3 that lab data evidence is important in determining if the current case of meningitis is being caused by the e.Coli organism. As pointed out in the section above, in other (higher level) knowledge

a. undertake a partial pattern match _only_.

Thus in MDX systems, reliance for all computations dealing with likelihoods is carried on a basis local to an individual diagnostic specialist, while in MYCIN computations dealing with likelihoods _must_ be undertaken on a global basis because individual production rules do not possess the power to "integrate" the results of the pattern match they embody.

3. Both diagnostic specialists and knowledge groups in MDX systems are _named_ entities, while in MYCIN the individual production rules are unnamed; the MDX approach employs the "principle of explicit naming" as put forward by Marr [6].

2.2. The Original Domains

With respect to the use of a global uncertainty calculus, the domain of MYCIN, i.e., infectious disease, contains for the most part knowledge which is "associational" or "statistical" in nature. Instead of having an underlying basis in known causal relations, the knowledge is derived from medical population studies.

On the other hand, the original MDX system dealt with the domain of cholestatic liver disease. Physicians in this domain rely heavily on laboratory, imaging, and clinical examination data. Much of the domain knowledge used in MDX could also be cast as "associational" data. But within MDX, a locally operating, concept-sensitive combination of likelihood has been found to be both powerful and robust.

need for another layer of intermediate abstractions by splitting the constituent variables into two or more subgroups, each corresponding to a meaningful and potentially relevant conceptualization.

Space here limits going into the issues of how to decide on the appropriate number of levels to represent each conceptualization, or how to insulate the decision at each level of abstraction from dependence on the purpose for which the higher-level abstraction or decision process may put it. The issues here have many points of contact with the principles that Marr [6] suggests should guide the processing of the enormous amount of low level visual information into abstraction structures at higher and higher levels.

With the brief introduction to MDX-style knowledge groups above, the following differences between the MDX paradigm and the MYCIN paradigm become more clear.

1. The smallest unit of domain knowledge constructed to deal with the combination of uncertain patient data in the MDX approach is the knowledge group, while in the MYCIN approach the smallest unit is the individual production rule.

2. The information processing task of the individual knowledge group following MDX is to

    a. undertake partial pattern matches against the patient specific data and

    b. produce a single abstract, symbolic measure of the contribution of domain knowledge of this knowledge group,

   while in MYCIN the information processing task of the individual production rule is to whatever knowledge group invoked the current one.

This method of combining uncertainties is quite similar to the "signature table" idea proposed by Samuel [5] several years ago in the design of checker playing programs. Each row of values for the constituent pieces of evidence can be regarded as a signature. Just as Samuel's proposed signature systems containing several layers to reduce the computational and conceptual complexity of each signature table, we also use sufficient number of layers to make each layer, or step in the abstraction, manageable both computationally and in terms of knowledge acquisition from experts.

The last point may need some elaboration. On looking at Figure 1a, the *first reaction might* be that the number of rows grows exponentially with the number of columns and the number of discrete values for each constituent. Thus the complexity of the knowledge acquisition process, i.e., the process of getting values for the last column from human experts, may seem to be forbidding.

There are two sources of relief from this complexity. First, it turns out that often each row need not be considered separately. For instance, for Figure 1c, a mechanic might say, "If the observational evidence for battery trouble is very strong, then the historical evidence does not matter. It is only when the observational evidence is somewhat weak that I consider the contribution of the historical evidence." This sort of dependency would reduce the number of rows to be independently filled in. Secondly, a large number of rows might suggest the

1. H. Sherman. A Comparitive Study of Computer-Aided Clinical Diagnosis of Birth Defects. Master Th., Dept of Electrical Engineering and Computer Science, MIT, 1981.

2. E. Shortliffe. Computer Based Medical Consultations: MYCIN. Elsevier North Holland Inc., 1976.

3. B. Chandrasekaran. Decomposition of Domain Knowledge into Knowledge Sources: The MDX Approach. Proc. 4th Nat. Conf. Canadian Society for Computational Studies of Intelligence, Saskatchewan, May, 1982.

4. B. Chandrasekaran, S. Mittal, and J. W. Smith. Reasoning with Uncertain Knowledge: The MDX Approach. Proc. 1st Ann. Joint Conf. of the AMIA, 1982.

5. A. Samuel. "Some studies in machine learning using the game of checkers." IBM Journal of Research and Development (1967), 601-617.

6. D. Marr. "Early Processing of Visual Information." Phil. Trans. Roy. Soc. 197(1976).

7. Mark Stefik, Danniel G. Bobrow, Sanjay Mittal, Lynn Conway. "Knowledge Programming in LOOPS." AI Magazine 4(1983), 3-13.

8. F. Gomez, B. Chandrasekaran. "Knowledge Organization and Distribution for Medical Diagnosis." IEEE Transactions on Systems, Man, and Cybernetics SMC-11, 1 (January 1981), 34-42.

Figure Captions

Figure 1a: Top Level Knowledge Group For "Battery"

Figure 1b: Low Level Knowledge Group For "Battery"

Figure 1c: Another Low Level Knowledge Group For "Battery"

Figure 2: An MDX-MYCIN Knowledge Group

Figure 3: Results From the Medical Literature Cases

Figure 4: MDX-MYCIN Hierarchy After Running MYCIN Case 232

Figure 5: Establishing Values For Specialists in Figure 4

Figure 6: StaphCoagPos Knowledge Groups

Figure 7: Values Returned By StaphCoagPos Knowledge Groups

Figure 8: Final Stage Of The Debugging Process

batteryProblem.TopLevel

results from observations	results from history	result from this Knowledge Group
= 3	DON'T CARE	+3
> 0	> 0	+2
DON'T CARE	> 0	+1
> 0	DON'T CARE	+1
< 0	DON'T CARE	-2

Fig 1a

batteryProblem.Observations

engine turns over slowly, then not at all	headlights dim	result from this Knowledge Group
TRUE	TRUE	+3
TRUE	DON'T CARE	+1
FALSE	FALSE	-2

Fig 1b

batteryProblem.History

season is Winter	batteryAge > 5 yrs	result from this Knowledge Group
TRUE	TRUE	+3
DON'T CARE	TRUE	+2
FALSE	FALSE	-1

Fig 1c

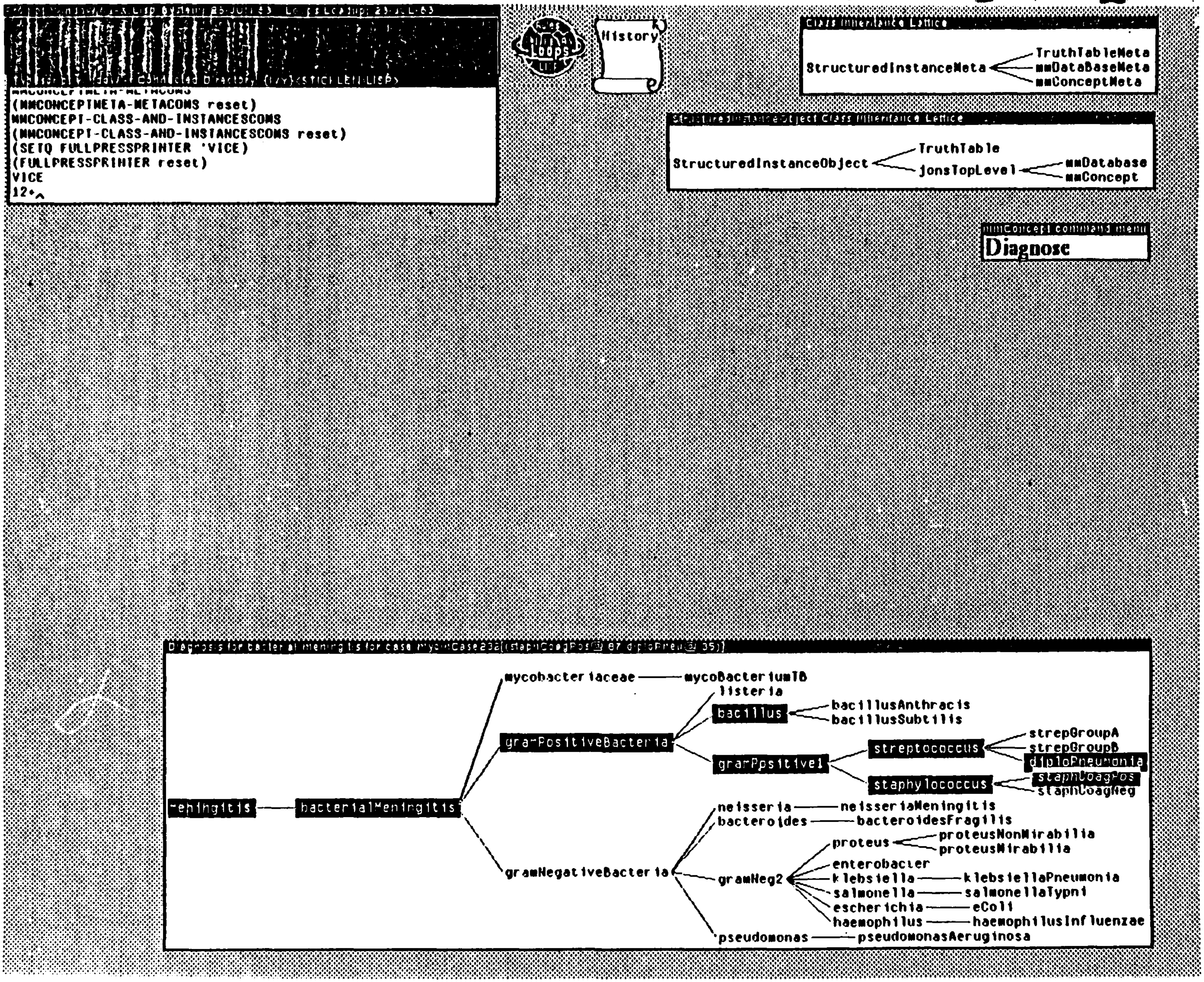
eColi.LAB

stain AND morphology known	cbc > 2.5	result from this Knowledge Group
TRUE	TRUE	+1
TRUE	FALSE	+3
FALSE	DON'T CARE	0

Fig 2

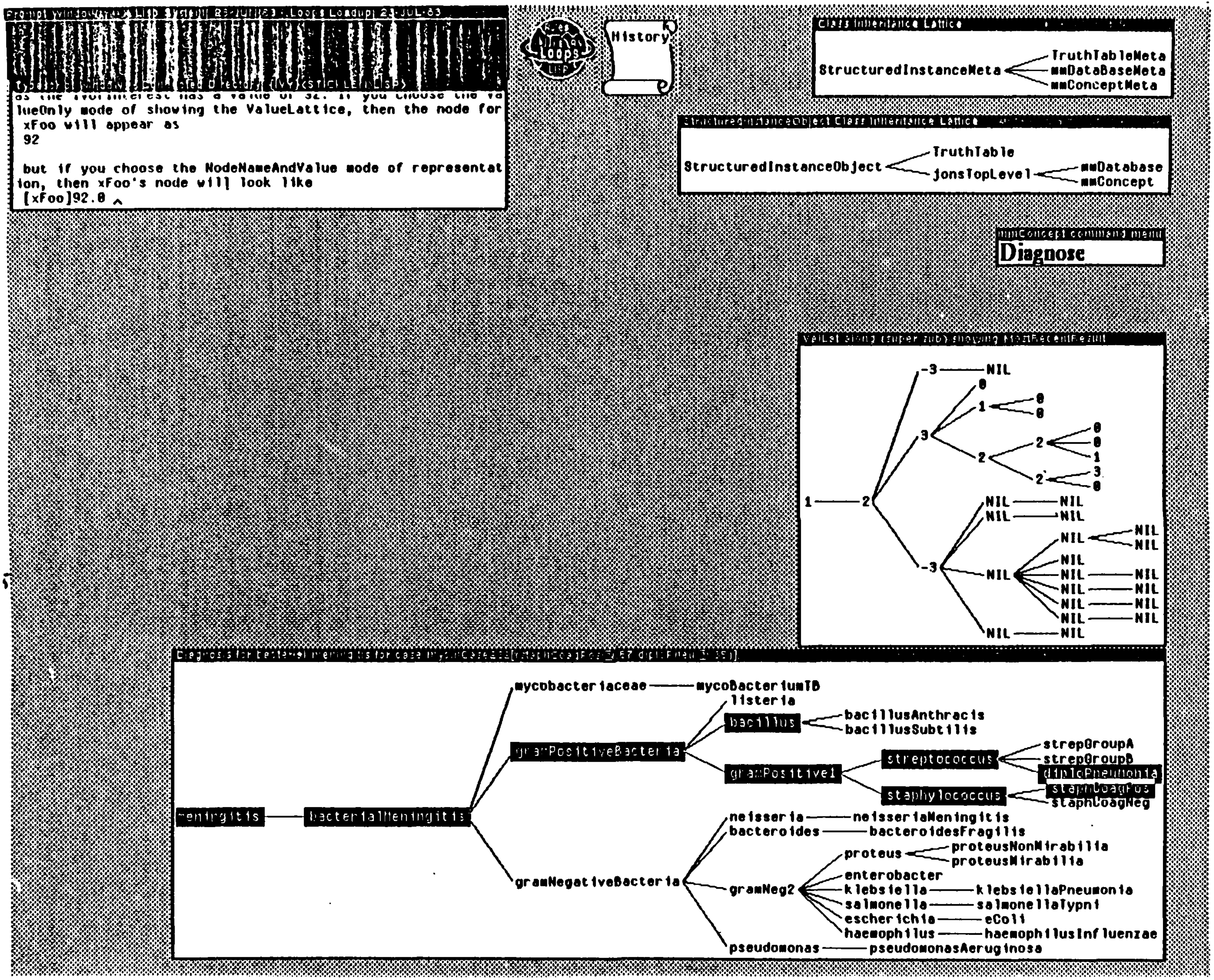
		MYCIN	MDX-MYCIN
FULL DATA	metric 1	1.0	1.0
	metric 2	0.8	0.3
INITIAL DATA	metric 1	0.9	1.0
	metric 2	0.9	2.6

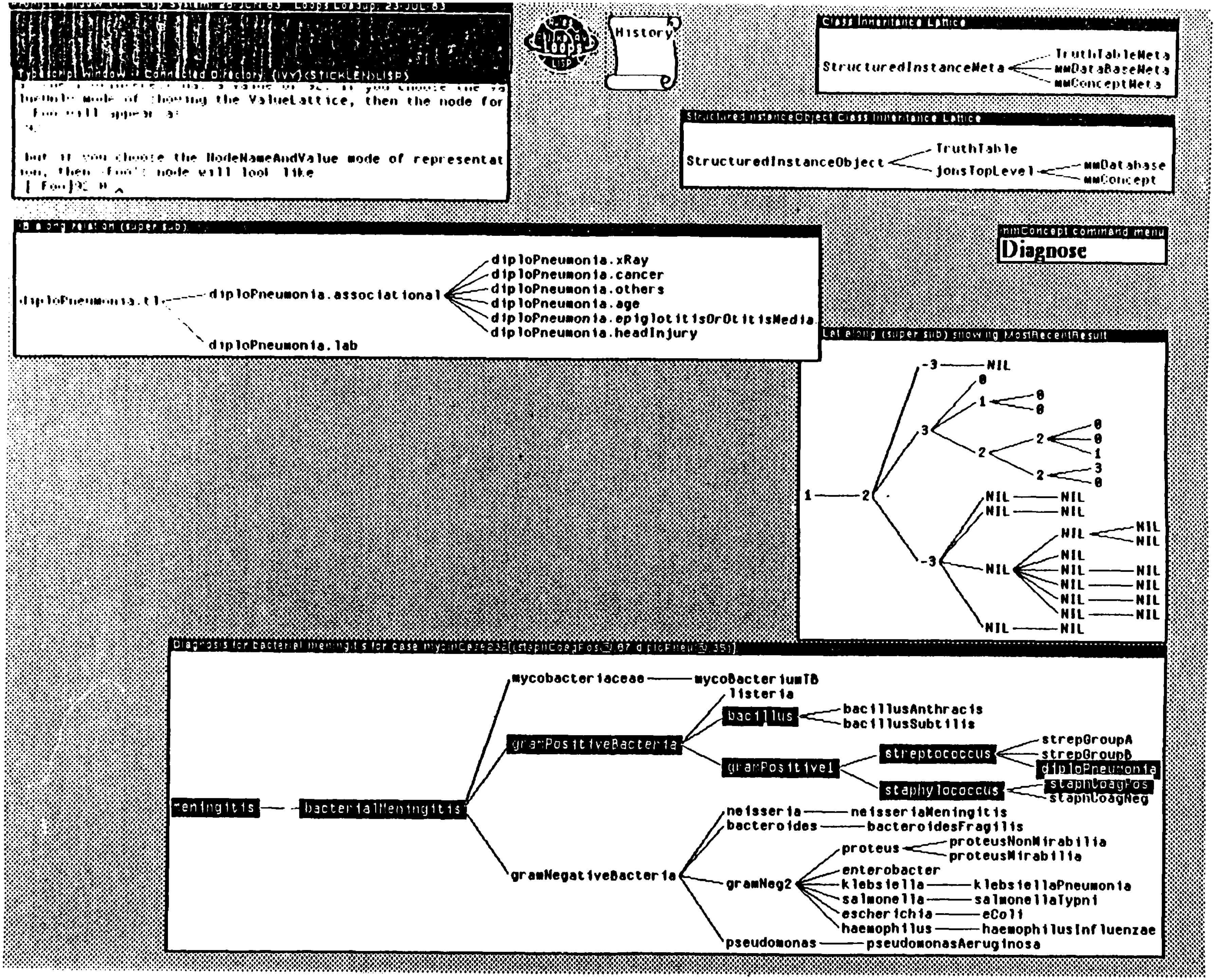
Fig 3



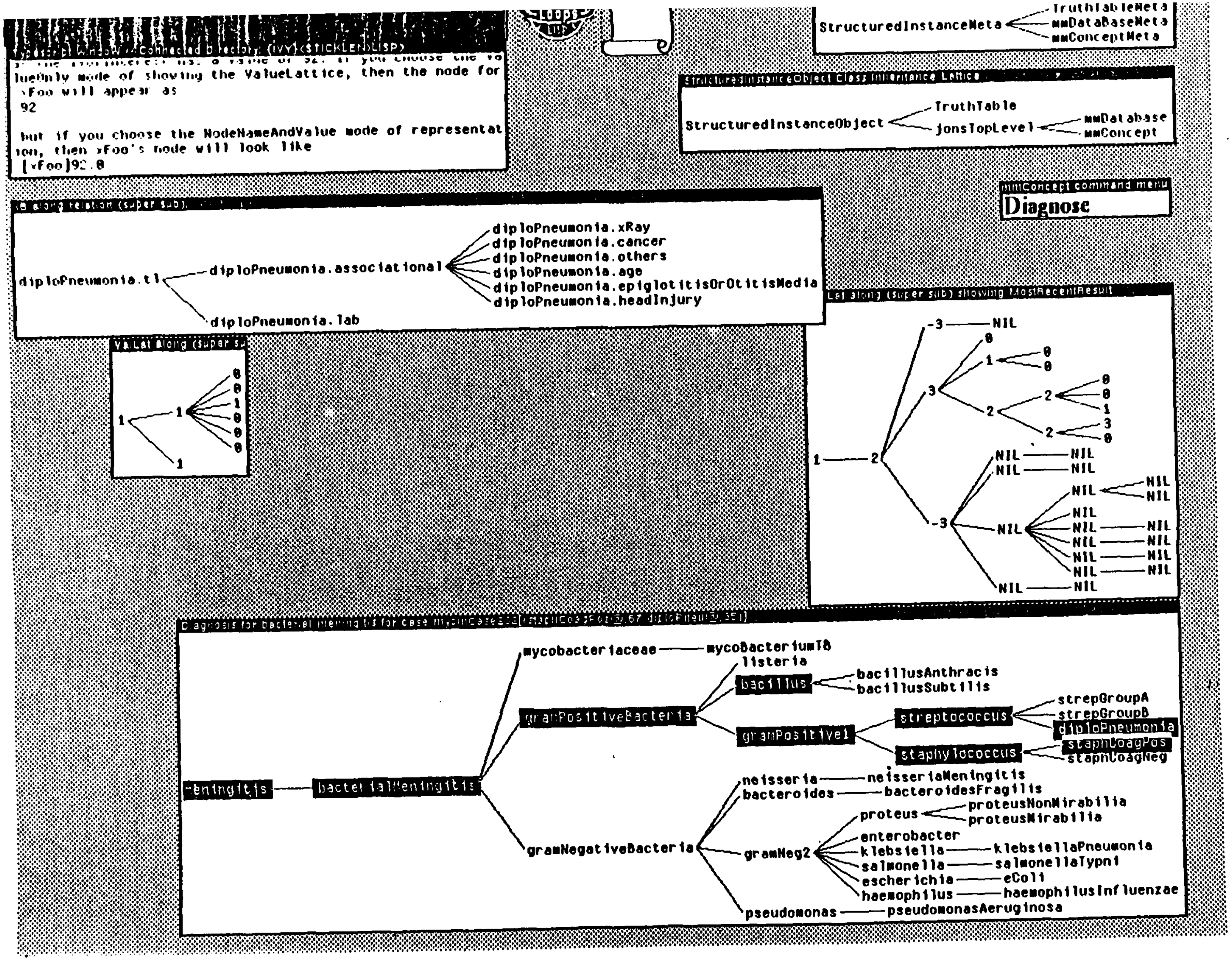
(MMCONCEPTMETA-METACOMS reset)
MMCONCEPT-CLASS-AND-INSTANCESCOMS
(MMCONCEPT-CLASS-AND-INSTANCESCOMS reset)
(SETQ FULLPRESSPRINTER 'VICE)
(FULLPRESSPRINTER reset)
VICE
12←
History
StructuredInstanceMeta
TruthTableMeta
mmDataBaseMeta
mmConceptMeta
StructuredInstanceObject
TruthTable
jonsTopLevel
mmDatabase
mmConcept
Diagnose
meningitis
bacterialMeningitis
mycobacteriaceae
mycoBacteriumTB
listeria
gram-PositiveBacteria
bacillus
bacillusAnthracis
bacillusSubtilis
gramPositive1
streptococcus
strepGroupA
strepGroupB
diploPneumonia
staphylococcus
staphCoagPos
staphCoagNeg
neisseria
neisseriaMeningitis
bacteroides
bacteroidesFragilis
gramNegativeBacteria
gramNeg2
proteus
proteusNonMirabilia
proteusMirabilia
enterobacter
klebsiella
klebsiellaPneumonia
salmonella
salmonellaTyphi
escherichia
eColi
haemophilus
haemophilusInfluenzae
pseudomonas
pseudomonasAeruginosa

4


History
Class Inheritance Lattice
StructuredInstanceMeta
TruthTableMeta
mmDataBaseMeta
mmConceptMeta
StructuredInstanceObject
TruthTable
jonsTopLevel
mmDatabase
mmConcept
mmConcept command menu
Diagnose
lueOnly mode of showing the ValueLattice, then the node for
xFoo will appear as
92
but if you choose the NodeNameAndValue mode of representat
ion, then xFoo's node will look like
[xFoo]92.0
meningitis
bacterialMeningitis
mycobacteriaceae
mycoBacteriumTB
listeria
bacillus
bacillusAnthracis
bacillusSubtilis
gramPositiveBacteria
gramPositive1
streptococcus
strepGroupA
strepGroupB
diploPneumonia
staphylococcus
staphCoagPos
staphCoagNeg
gramNegativeBacteria
neisseria
neisseriaMeningitis
bacteroides
bacteroidesFragilis
gramNeg2
proteus
proteusNonMirabilia
proteusMirabilia
enterobacter
klebsiella
klebsiellaPneumonia
salmonella
salmonellaTyphi
escherichia
eColi
haemophilus
haemophilusInfluenzae
pseudomonas
pseudomonasAeruginosa

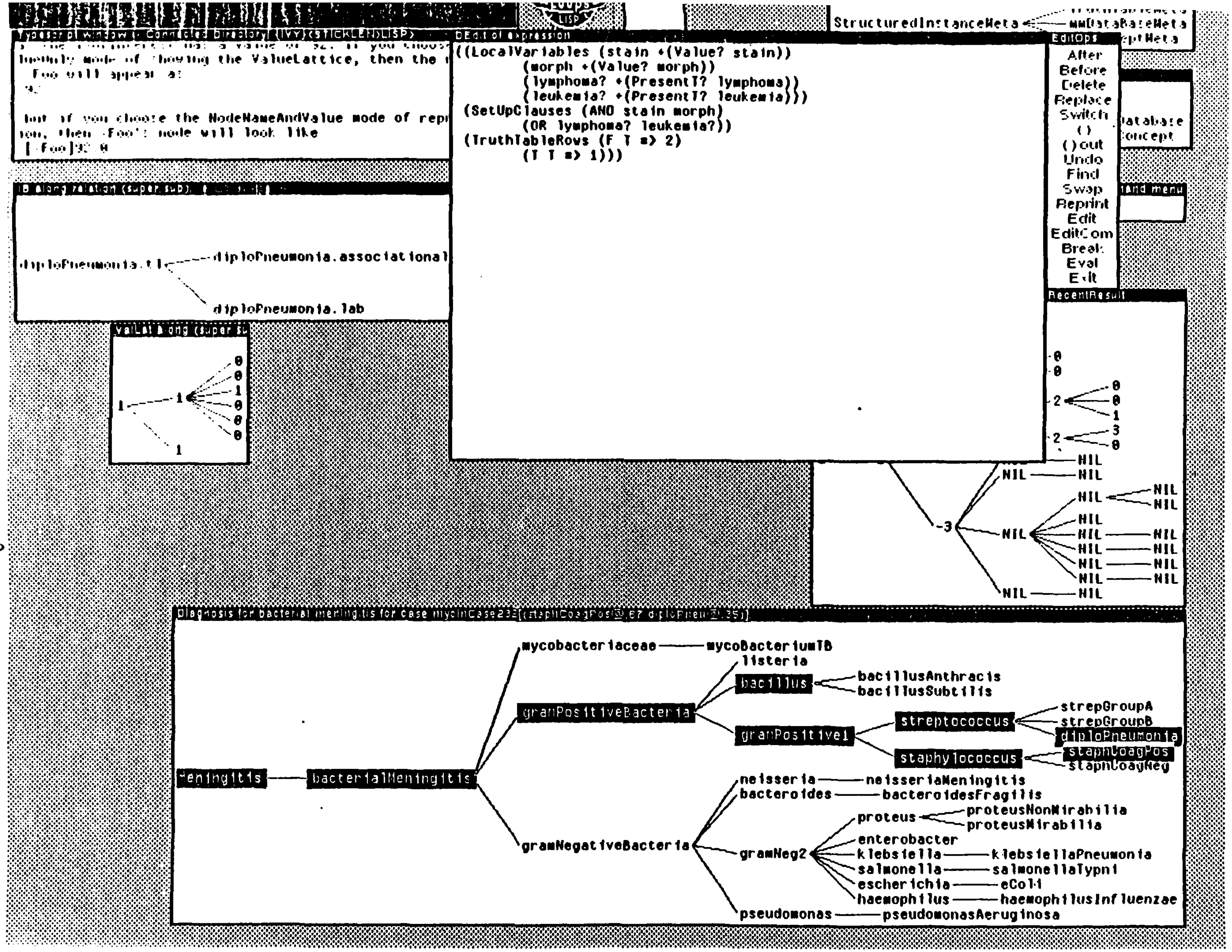


History
Class Inheritance Lattice
StructuredInstanceMeta
TruthTableMeta
mmDataBaseMeta
mmConceptMeta
StructuredInstanceObject Class Inheritance Lattice
StructuredInstanceObject
TruthTable
jonsTopLevel
mmDatabase
mmConcept
mmConcept command menu
Diagnose
but if you choose the NodeNameAndValue mode of representat
diploPneumonia.associational
diploPneumonia.xRay
diploPneumonia.cancer
diploPneumonia.others
diploPneumonia.age
diploPneumonia.epiglotitisOrOtitisMedia
diploPneumonia.headInjury
diploPneumonia.lab
Lattice (super sub) showing MostRecentResult
meningitis
bacterialMeningitis
mycobacteriaceae
mycoBacteriumTB
listeria
bacillus
bacillusAnthracis
bacillusSubtilis
gramPositiveBacteria
gramPositive1
streptococcus
strepGroupA
strepGroupB
diploPneumonia
staphylococcus
staphCoagPos
staphCoagNeg
neisseria
neisseriaMeningitis
bacteroides
bacteroidesFragilis
gramNegativeBacteria
gramNeg2
proteus
proteusNonMirabilia
proteusMirabilia
enterobacter
klebsiella
klebsiellaPneumonia
salmonella
salmonellaTyphi
escherichia
eColi
haemophilus
haemophilusInfluenzae
pseudomonas
pseudomonasAeruginosa



lueOnly mode of showing the ValueLattice, then the node for
xFoo will appear as
92
but if you choose the NodeNameAndValue mode of representat
ion, then xFoo's node will look like
[xFoo]92.0
StructuredInstanceMeta
TruthTableMeta
mmDataBaseMeta
mmConceptMeta
StructuredInstanceObject Class Inheritance Lattice
StructuredInstanceObject
TruthTable
jonsTopLevel
mmDatabase
mmConcept
mmConcept command menu
Diagnose
diploPneumonia.t1
diploPneumonia.associational
diploPneumonia.lab
diploPneumonia.xRay
diploPneumonia.cancer
diploPneumonia.others
diploPneumonia.age
diploPneumonia.epiglotitisOrOtitisMedia
diploPneumonia.headInjury
Lat along (super sub) showing MostRecentResult
Diagnosis for bacterial meningitis for case
meningitis
bacterialMeningitis
mycobacteriaceae
mycoBacteriumTB
gramPositiveBacteria
listeria
bacillus
bacillusAnthracis
bacillusSubtilis
gramPositive1
streptococcus
strepGroupA
strepGroupB
diploPneumonia
staphylococcus
staphCoagPos
staphCoagNeg
gramNegativeBacteria
neisseria
neisseriaMeningitis
bacteroides
bacteroidesFragilis
gramNeg2
proteus
proteusNonMirabilia
proteusMirabilia
enterobacter
klebsiella
klebsiellaPneumonia
salmonella
salmonellaTypni
escherichia
eColi
haemophilus
haemophilusInfluenzae
pseudomonas
pseudomonasAeruginosa

fig 1


StructuredInstanceMeta
Typescript window
BEdit of expression
((LocalVariables (stain ←(Value? stain))
(morph ←(Value? morph))
(lymphoma? ←(PresentT? lymphoma))
(leukemia? ←(PresentT? leukemia)))
(SetUpClauses (AND stain morph)
(OR lymphoma? leukemia?))
(TruthTableRows (F T => 2)
(T T => 1)))
EditOps
After
Before
Delete
Replace
Switch
()
()out
Undo
Find
Swap
Reprint
Edit
EditCom
Break
Eval
Exit
RecentResult
Database
Concept
the ValueLattice, then the
Foo will appear as
but if you choose the NodeNameAndValue mode of rep
then Foo's node will look like
diploPneumonia.associational
diploPneumonia.lab
Diagnosis for bacterial meningitis for case mycinCase23
meningitis
bacterialMeningitis
mycobacteriaceae
mycoBacteriumTB
gramPositiveBacteria
listeria
bacillus
bacillusAnthracis
bacillusSubtilis
gramPositive1
streptococcus
strepGroupA
strepGroupB
diploPneumonia
staphylococcus
staphCoagPos
staphCoagNeg
gramNegativeBacteria
neisseria
neisseriaMeningitis
bacteroides
bacteroidesFragilis
gramNeg2
proteus
proteusNonMirabilia
proteusMirabilia
enterobacter
klebsiella
klebsiellaPneumonia
salmonella
salmonellaTyphi
escherichia
eColi
haemophilus
haemophilusInfluenzae
pseudomonas
pseudomonasAeruginosa

APPENDIX 2

To appear in the Special Issue of Intn'l Jrnl. of Computers and Mathematics on "practical artificial intelligence systems."

CSRL: A Language for Expert Systems for Diagnosis*

Tom Bylander, Sanjay Mittal**, and B. Chandrasekaran
Artificial Intelligence Group
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210 USA

Abstract

We present CSRL (Conceptual Structures Representation Language) as a language to facilitate the development of expert diagnosis systems based on a paradigm of "cooperating diagnostic specialists." In our approach, diagnostic reasoning is one of several generic tasks, each of which calls for a particular organizational and problem solving structure. A diagnostic structure is composed of a collection of specialists, each of which corresponds to a potential hypothesis about the current case. They are organized as a classification or diagnostic hierarchy, e.g., a classification of diseases. A top-down strategy called establish-refine is used in which either a specialist establishes and then refines itself, or the specialist rejects itself, pruning the hierarchy that it heads. CSRL is a language for representing the specialists of a diagnostic hierarchy and the diagnostic knowledge within them. The diagnostic knowledge is encoded at various levels of abstractions: message procedures, which describe the specialist's behavior in response to messages from other specialists; knowledge groups, which determine how data relate to features of the hypothesis; and rule-like knowledge, which is contained within knowledge groups.

---

*This an expanded version of a paper of the same title which was presented at the 1983 International Joint Conference on Artificial Intelligence

**Currently at Knowledge Systems Area, Xerox PARC, 3333 Coyote Hill Rd., Palo Alto, CA 94304 USA

CSRL: A Language for Expert Systems for Diagnosis*

Tom Bylander, Sanjay Mittal**, and B. Chandrasekaran
Artificial Intelligence Group
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210 USA

1 Introduction

Many kinds of problem solving for expert systems have been proposed within the AI community. Whatever the approach, there is a need to acquire the knowledge in a given domain and implement it in the spirit of the problem solving paradigm. Reducing the time to implement a system usually involves the creation of a high level language which reflects the intended method of problem solving. For example, EMYCIN [1] was created for building systems based on MYCIN-like problem solving [2]. Such languages are also intended to speed up the knowledge acquisition process by allowing domain experts to input knowledge in a form close to their conceptual level. Another goal is to make it easier to enforce consistency between the expert's knowledge and its implementation.

CSRL (Conceptual Structures Representation Language) is a language for implementing expert diagnostic systems that are based on our approach to

---

*This an expanded version of a paper of the same title which was presented at the 1983 International Joint Conference on Artificial Intelligence.

**Currently at Knowledge Systems Area, Xerox PARC, 3333 Coyote Hill Rd., Palo Alto, CA 94304 USA

The Red expert system is composed of three major subsystems, one of which is implemented in CSRL. The non-CSRL subsystems are a data base which maintains and answers questions about reaction records (reactions of the patient's blood in selected blood samples under a variety of conditions), and a overview system, which assembles a composite hypothesis of the antibodies that would best explain the reaction record [13]. CSRL is used to implement specialists corresponding to each antibody that Red knows about (about 30 of the most common ones) and to each antibody subtype (different ways that the antibody can react).

The major function of the specialists is to rule out antibodies and their subtypes whenever possible, thus simplifying the job of the overview subsystem, and to assign confidence values, informing overview of which antibodies appear to be more plausible. The *specialists query* the data base for information about the test reactions and other patient information, and also tell the data base to perform certain operations on reaction records.

An interesting feature of Red is how it handles the problem of interacting hypotheses. It is possible for the patient's blood to have practically any number or combination of antibodies, making it very hard for a single specialist to determine how well it will fit with other specialists in a composite hypothesis. In Red, each specialist is encoded to assume that it is independent — it looks at the data as if no other specialist can account for the same data. The knowledge of how the specialists can interact is left to the overview subsystem. This would be problematic if few specialists could rule themselves out, but so happens that in this domain, it is rare to have more than a few antibodies that cannot be independently ruled out. Thus Red's CSRL subsystem makes overview's problem solving computationally feasible since initial data from the user. This includes the major symptom that the user notices (such as stalling) and the situation when this occurs (e.g., accelerating and cold engine temperature). Any additional questions are asked while Auto-Mech's specialists are running. The diagnosis then starts and continues until the user is satisfied that the diagnosis is complete. The user must make this decision since the data that Auto-Mech uses are very weak at indicating specific problems and, more importantly, Auto-Mech is unable to make the repair and determine whether the problem has been fixed.

A major part of Auto-Mech's development was determining the assumptions that would be made about the design of the automobile engine and the data that the program would be using. Different automobile engine designs have a significant effect on the hypotheses that are considered. A carbureted engine, for example, will have a different set of problems than a fuel injected engine (the former can have a broken carburetor). The data was assumed to come from commonly available resources. The variety of computer analysis information that is available to mechanics today was not considered in order to simplify building Auto-Mech.

4.2 Red

Red is an expert system whose domain is red blood cell antibody identification [12]. An everyday problem that a blood bank contends with is the selection of units of blood for transfusion during major surgery. The primary difficulty is that antibodies in the patient's blood may attack the foreign blood, rendering the new blood useless as well as presenting additional danger to the patient. Thus identifying the patient's antibodies and selecting blood which will not react with them is a critical task for nearly all red blood transfusions.

3.5 The CSRL Environment

The current version of CSRL is implemented in INTERLISP-D and LOOPS, an object-oriented programming tool. Each specialist is implemented as a LOOPS class, which is instantiated for each case that is run. The LOOPS class hierarchy is used to specify default message procedures and shared knowledge groups, making it easy to encode a default establish-refine strategy, and letting the user incrementally modify this strategy and add strategies as desired. A graphical interface displays the specialist hierarchy, and through the use of a mouse, allows the user to easily access and modify any part of the hierarchy. Additional facilities for debugging and explanation are being implemented.

4 Expert Systems that use CSRL

4.1 Auto-Mech

Auto-Mech is an expert system which diagnoses fuel problems in automobile engines [6]. This domain was chosen to demonstrate the viability of our approach to non-medical domains, as well as to gain experience and feedback on CSRL.* The purpose of the fuel system is to deliver a mixture of fuel and air to the air cylinders of the engine. It can be divided into major subsystems (fuel delivery, air intake, carburetor, vacuum manifold) which correspond to initial hypotheses about fuel system faults.

Auto-Mech consists of 34 CSRL specialists in a hierarchy which varies from four to six levels deep. Its problem solving closely follows the establish-refine strategy. Before this strategy is invoked, Auto-Mech collects some

*Auto-Mech was developed using an early version of the language.

the values of the relevant and gas knowledge group (the latter queries the user about the temporal relationship between the onset of the problem and when gas was last bought). In this case, if the value of the relevant knowledge group is 3 and the value of the gas knowledge group is greater then or equal to 0, then the value of the summary knowledge group (and consequently the confidence value of BadFuel) is 3, indicating that a bad fuel problem is very likely.

PUT FIGURE 6 HERE

3.4 Comparison with Rule-Based Languages

There is nothing in CSRL that is not programmable within rule-based languages such as OPS5 [10] or EMYCIN [1]. The difference between CSRL and these languages is that CSRL makes a commitment to a particular organizational and programming style. CSRL is not intended to be a general purpose representation language, but is built specifically for the classificatory diagnosis problem. It is possible to program in a rule-based language so that there is an implicit relationship between rules so that they correspond to knowledge groups and specialists. R1, although not a diagnostic expert system, is an excellent example of how one creates implicit grouping of rules in such a system [11]. The central idea underlying CSRL is to make these relationships explicit. The expert system implementor is then relieved from trying to impose an organization on a organization-less system and is free to concentrate on the conceptual structure of the domain. Also, there is a greater potential to embed explanation and debugging facilities which can take advantage of the expert system organization.

of the features.* By examining the results of test cases, the knowledge groups are relatively easy to debug since the attention of the domain expert can be directed to the specific area of knowledge which derived the incorrect result.

As an example, figure 5 is the relevant knowledge group of the BadFuel specialist mentioned above. It determines whether the symptoms of the automobile are consistent with bad fuel problems. The expressions query the user (who is the data base for Auto-Mech) for whether the car is slow to respond, starts hard, has knocking or pinging sounds, or has the problem when accelerating. "AskYNU?" is a LISP function which asks the user for a Y, N, or U (unknown) answer from the user, and translates the answer into T, F, or U, the values of CSRL's three-valued logic. Each set of tests in the if-then part of the knowledge group is evaluated until one matches. The value corresponding to this "rule" becomes the value of the knowledge group. For example, the first rule tests whether the first expression is true (the "?" means doesn't matter). If so, then -3 becomes the value of the knowledge group. Otherwise, other rules are evaluated. The value of the knowledge group will be 1 if no rule matches. This knowledge group encodes the following diagnostic knowledge:

> If the car is slow to respond or if the car starts hard, then BadFuel is not relevant in this case. Otherwise, if there are knocking or pinging sounds and if the problem occurs while accelerating, then BadFuel is highly relevant. In all other cases, BadFuel is only mildly relevant.

PUT FIGURE 5 HERE

Figure 6 is the summary knowledge group of BadFuel. Its expressions are

---

*Actually, any number of knowledge group levels can be implemented.

parameterized and message procedures can declare local variables.

3.3 Knowledge Groups

The kgs section of a specialist definition contains a list of knowledge groups, which are used to evaluate how selected data indicate various features or intermediate hypotheses that relate to specialist's hypothesis. A knowledge group can be thought of as a cluster of production rules which map the values of a list of expressions (boolean and arithmetic operations on data) to some conclusion on a discrete, symbolic scale. Different types of knowledge groups perform this mapping differently, e.g., directly mapping values to conclusions, or having each rule add or subtract a set number of "confidence" units.

Knowledge groups are intended for encoding the heuristics that a domain expert uses for inferring features of a hypothesis from the case description. The main problem is that this inference is uncertain -- there is rarely a one-to-one mapping from data to the features of the hypothesis. The way that this is handled in CSRL is borrowed from the uncertainty handling techniques used in MDX [9].

Each feature or intermediate hypothesis is associated with a knowledge group. The data that the domain expert uses to evaluate the feature is encoded as expressions in the knowledge group. These are usually queries to a separate data base system. Each combination of values of the expressions is then mapped to a level of confidence as determined by the domain expert. This set of knowledge groups becomes the data for another knowledge group, which determines the confidence value of the specialist from the confidence values is set to the value of the relevant knowledge group. In CSRL, a confidence value scale of -3 to +3 is used (integers only). A value of +2 or +3 indicates that the specialist is established. In this case, the procedure corresponds to the following diagnostic knowledge.

> First perform a preliminary check to make sure that BadFuel is a relevant hypothesis to hold. If it is not (the relevant knowledge group is less than 0), then set BadFuel's confidence value to the degree of relevancy. Otherwise, perform more complicated reasoning (the summary knowledge group combines the values of other knowledge groups) to determine BadFuel's confidence value.

PUT FIGURE 3 HERE

Figure 4 shows a Refine procedure which is a simplified version of the one that BadFuel uses. "subspecialists" is a keyword which refers to the subspecialists of the current specialist. The procedure calls each subspecialist with an Establish message.* If the subspecialist establishes itself (+? tests if the confidence value is +2 or +3), then send it a Refine message.

PUT FIGURE 4 HERE

CSRL has a variety of other kinds of statements and expressions so that more complicated strategies can be implemented. For example, a "Reset" statement deletes the confidence value and the knowledge group values of a specialist. This might be used when additional tests are performed, making it necessary to recalculate the confidence value. Also, messages can be

---

*For convenience, many of CSRL's control constructs mimic those of INTERLISP; however, these constructs are executed by the CSRL interpreter, not by using LISP EVAL. LISP code is allowed within message procedures, but only by within a construct called "DoLisp". This is not intended to let specialists have arbitrary code, but to allow interaction with other LISP-implemented systems.

3.2 Message Procedures

The messages section of a specialist contains a list of message procedures, which specify how the specialist will respond to different messages from its superspecialist.* "Establish", "Refine", "Establish-Refine" (combines Establish and Refine), and "Suggest" are predefined messages in CSRL; additional messages may be defined by the user. Below, we will examine how Establish and Refine procedures are typically constructed.

Message procedures are the highest level of abstraction for diagnostic knowledge within specialists. Just as in general message passing languages, messages provide a way to invoke a particular kind of response without having to know what procedure to invoke. Strategies for diagnosis, such as establish-refine, are usually easy to translate into a message protocol. However, CSRL does not provide any way to specify and enforce message protocols.

Figure 3 illustrates the Establish message procedure of the BadFuel specialist. "relevant" and "summary" are names of knowledge groups of BadFuel. "self" is a keyword which refers to the name of the specialist. This procedure first tests the value of the relevant knowledge group. (If this knowledge group has not already been executed, it is automatically executed at this point.) If it is greater than or equal to 0, then BadFuel's confidence value is set to the value of the summary knowledge group, else it

---

*A specialist is not allowed to send messages to its superspecialist. However, other message passing routes are allowed. Specifically, a specialist may send a message to itself, across the hierarchy, and to indirect subspecialists. In the latter case, each interconnecting specialist is sent a "Suggest" message and decides within its Suggest message procedure whether or not to pass the original message downwards.

can "summarize" the results of several others. Knowledge groups are composed of rule-like knowledge which match the data against specific patterns, and when successful, provide values to be processed by the knowledge group.

3.1 Specialists

In CSRL, a diagnostic expert system is implemented by individually defining each specialist. The super- and subspecialists of the specialist are declared within the definition. Figure 2 is a skeleton of a specialist definition for the Bad Fuel node from figure 1. The declare section specifies its relationships to other specialists. The other sections of the specialist are examined below.

PUT FIGURE 2 HERE

Since CSRL is designed to use only a simple classification tree, many choices concerning the composition of the hierarchy must be made. This is a pragmatic decision, rather than a search for the "perfect" classification tree. The main criteria for evaluating a classification is whether enough evidence is normally available to make confident decisions. To decompose a specialist into its subspecialists, the simplest method is to ask the domain expert what subhypotheses should be considered next. Usually the subspecialists will differ from one another based on a single attribute (e.g., location, cause). For further discussion on this and other design decisions in CSRL, see Bylander and Smith [8].

issues and integrating their solutions into the diagnostic framework are problems for future research.

2.3 Differences from other Approaches

The usual approach to building knowledge based systems is to emphasize a general knowledge representation structure and different problem solvers which use that knowledge. One difference in this approach is that the organization of knowledge is not intended as a general representation for all problems. Rather it is tuned specifically for diagnosis. By limiting the type of problem to be solved, a specific organizational technique (classification hierarchy) and problem solving strategy (establish-refine) can be used to provide focus and control in the problem solving process.

Another difference is that the specialists in the hierarchy are not a static collection of knowledge. The knowledge of how to establish or reject is embedded within the specialists. Each specialist can then be viewed as a individual problem solver with its own knowledge base. The entire collection of specialists engages in distributed problem-solving.

3 CSRL

CSRL is a language for representing the specialists of a diagnostic hierarchy and the diagnostic knowledge within them. The diagnostic knowledge is encoded at various levels of abstractions. Message procedures describe the specialist's behavior in response to messages from other specialists. These contain the knowledge about how to establish or refine a specialist. Knowledge groups determine how selected data relate to various features or intermediate hypotheses that are related to the specialist. The selected data may be the values of other knowledge groups, so that a single knowledge group the specialist is eliminated from consideration. Otherwise the specialist suspends itself, and may later refine itself if its superior requests it.

With regard to figure 1, the following scenario might occur. First, the fuel system specialist is invoked, since it is the top specialist in the hierarchy. This specialist is then established, and the two specialists below it are invoked. Bad fuel problems is rejected, eliminating the three subspecialists of bad fuel from consideration. Finally, the fuel mixture specialist is established, and its subspecialists (not shown) are invoked.

An important companion to the diagnostic hierarchy is an intelligent data base assistant which organizes the case description, answers queries from the diagnostic specialists, and makes simple inferences from the data [7]. For example, the data base should be able to infer that the fuel tank is not empty if the car can be started. The diagnostic specialists are then relieved from knowing all the ways that a particular datum could be inferred from other data.

There are several issues relevant to diagnostic problem solving which we will not address here. The simple description above does not employ strategies for bypassing the hierarchical structure for common malfunctions, for handling multiple interacting hypothesis, or for accounting of the manifestations. Also, additional control strategies are required when many nodes are in a suspended state. For discussion on some of these topics, see Gomez and Chandrasekaran [5]. Test ordering, causal explanation of findings, and therapeutic action do not directly fall within the auspices of the classificatory diagnosis as defined here, but expertise in any of these areas would certainly enhance a diagnostic system. Fully resolving all of these

2.2 The Diagnostic Task

The diagnostic task is the identification of a case description with a specific node in a pre-determined diagnostic hierarchy. Each node in the hierarchy corresponds to a hypothesis about the current case. Nodes higher in the hierarchy represent more general hypotheses, while lower nodes are more specific. Typically, a diagnostic hierarchy is a classification of malfunctions of some object, and the case description contains the manifestations and background information about the object. For example, the Auto-Mech expert system [6] attempts to classify data concerning an automobile into a diagnostic hierarchy of fuel system malfunctions. Figure 1 illustrates a fragment of Auto-Mech's hierarchy. The most general node, the fuel system in this example, is the head node of hierarchy. More specific fuel system malfunctions such as fuel delivery problems are classified within the hierarchy.

PUT FIGURE 1 HERE

Each node in the hierarchy is associated with a _specialist_ which contains the diagnostic knowledge to evaluate the plausibility of the hypothesis from the case description. From this knowledge, the specialist determines a confidence value representing the amount of belief in the hypothesis. If this value is high enough, the specialist is said to be _established_.

The basic strategy of the diagnostic task is a process of hypothesis refinement, which we call _establish-refine_. In this strategy, if a specialist establishes itself, then it _refines_ the hypothesis by invoking its subspecialists, which also perform the establish-refine strategy. If its confidence value is low, the specialist _rejects_ the hypothesis, and performs no further actions. Note that when this happens, the whole hierarchy below elements as well.) Other examples include knowledge-directed data retrieval, consequence finding, and a restricted form of design.

Each generic task calls for a particular organizational and problem solving structure. Given a specific kind of task to perform, the idea is that specific ways to organize and use knowledge are ideally suited for that task.

Even when the specification of a problem is reduced to a given task within a given domain, the amount of knowledge which is needed can still be enormous (e.g., diagnosis in medicine). In our approach, the knowledge structure for a given task and domain is composed of specialists, each of which specialize in different concepts of the domain. Domain knowledge is distributed across the specialists, dividing the problem into more manageable parts, and organizing the knowledge into chunks which become relevant when the corresponding concepts become relevant during the problem solving.

Decomposing a domain into specialists raises the problem of how they will coordinate during the problem solving process. First, the specialists as a whole are organized, primarily around the "subspecialist-of" relationship. Each task may specify additional relationships that may hold between specialists. Second, each task is associated with a set of strategies which take advantage of these relationships and the problem solving capabilities of the individual specialists. The choice of what strategy to follow is not a global decision, but chosen by the specialists during problem solving.

diagnostic problem solving. This approach is an outgrowth of our group's experience with MDX, a medical diagnostic program [3], and with applying MDX-like problem solving to other medical and non-medical domains. CSRL facilitates the development of diagnostic systems by supporting constructs which represent diagnostic knowledge at appropriate levels of abstraction.

First, we will overview the relationship of CSRL to our overall theory of problem solving types and the diagnostic problem solving that underlies CSRL. We then present CSRL, illustrating how its constructs are used to encode diagnostic knowledge. Two expert systems under development in our laboratory which use CSRL are then briefly described. Based on our experience with these systems, we point out where improvements in CSRL are needed.

2 Classificatory Diagnosis

The central problem solving of diagnosis, in our view, is classificatory activity. This is a specific type of problem solving in our approach, meaning that a special kind of organization and special strategies are strongly associated with performing expert diagnosis. In this section, we will briefly review the theory of problem solving types as presented by Chandrasekaran [4], and the structure and strategies of the diagnostic task [5].

2.1 Types of Problem Solving

We propose that expert problem solving is composed of a collection of different problem solving abilities. The AI group at Ohio State has been working at identifying well-defined types of problem solving (called generic tasks), one of which is classificatory diagnosis. (For the purposes of this discussion, we will use "diagnosis" in place of "classificatory diagnosis" with the understanding that the complete diagnostic process includes other it considerably reduces the amount of search that would otherwise be necessary.

5 Needed Improvements in CSRL

The largest flaw in CSRL is that there is no strategy that determines when diagnosis should stop. Currently, the default procedures simply ask the user if the current diagnosis is satisfactory. Some notion of what it means to account for the data needs to be added to the language. The work on Red's overview system is a step in this direction, but there needs to be more integration of overview and CSRL (currently overview starts after the specialists are finished), and a better understanding of what kinds of interactions can occur between two hypotheses. Progress in this area would also help increase the focus of the diagnosis, i.e., the diagnosis could concentrate on accounting for the most important manifestation(s).

Another problem is the meaning of the confidence value of a specialist. In MDX, this value was directly associated with the amount of belief in the specialist. However in both Auto-Mech and Red, this meaning had to be slightly altered to fit the purposes of the expert system. In Auto-Mech the confidence value is used to indicate whether the hypothesis was worth pursuing. In Red it is used to indicate the specialist's plausibility given the independence assumption mentioned earlier. It is not possible in either expert system to confirm a specialist without outside help. In Auto-Mech a repair or highly specific test must be performed while in Red all the specialists must be considered together. This does not create a problem for the process of establish-refine problem solving, but makes it difficult to explain what the confidence value means. Any explanation facility must understand the assumptions that are being made to make coherent explanations.

6 Conclusion

We believe that the development of complex expert systems will depend on the availability of special purpose languages with organizational and problem-solving tools that match the conceptual structure of the domain. CSRL represents an initial step in this direction. It provides facilities to organize diagnostic knowledge in accordance with the structure of the domain. In particular, CSRL's constructs facilitate th· encoding of rule-like and strategic knowledge into appropriate abstractions: knowledge groups, message procedures, and specialists.

ACKNOWLEDGMENTS

We would like to acknowledge Jack Smith and Jon Sticklen for many fruitful discussions concerning CSRL's design. Many improvements in the language are due to Mike Tanner and John Josephson, who implemented the CSRL specialists in Auto-Mech and Red. The language development is funded by a grant from the Battelle Memorial Laboratories University Distribution Program, and experimentation and application in different domains is supported by AFOSR grant 82-0255, and NSF grant MCS-8103480.

REFERENCES

1. W. van Melle. A Domain Independent Production-Rule System for Consultation Programs. Proc. Sixth Interational Conf. on Artificial Intelligence, Tokyo, 1979, pp. 923-925.

2. E. H. Shortliffe. Computer-Based Medical Consultations: MYCIN. Elsevier, New York, 1976.

3. B. Chandrasekaran and S. Mittal. Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems. In Advances in Computers, Academic Press, New York, 1983, pp. 217-293.

4. B. Chandrasekaran. "Towards a Taxonomy of Problem Solving Types." AI Magazine 4, 1 (Winter/Spring 1983), 9-17.

5. F. Gomez and B. Chandrasekaran. "Knowledge Organization and Distribution for Medical Diagnosis." IEEE Trans. Systems, Man and Cybernetics SMC-11, 1 (January 1981), 34-42.

6. M. C. Tanner and T. Bylander. Application of the CSRL Language to the Design of Expert Diagnosis Systems: The Auto-Mech Experience. to appear in Proc. of the Joint Services Workshop on Artificial Intelligence in Maintenence, 1984.

7. S. Mittal and B. Chandrasekaran. "Conceptual Representation of Patient Data Bases." J. of Medical Systems 4, 2 (1980), 169-185.

8. T. Bylander, and J. W. Smith. Using CSRL for Medical Diagnosis. Proc. Second International Conf. on Medical Computer Science and Computational Medicine, IEEEComputerSociety, Glouster, Ohio, 1983.

9. B. Chandrasekaran, S. Mittal, and J. W. Smith. Reasoning with Uncertain Knowledge: The MDX Approach. Proc. Congress of American Medical Infomatics Association, San Francisco, 1982.

10. C. L. Forgy. OPS5 Users Manual. Tech. Rept. CMU-CS-81-135, Carnegie-Mellon University, 1981.

11. J. McDermott. "R1: A Rule-based Configurer of Computer Systems." Artificial Intelligence 19, 1 (1982), 39-88.

12. J. W. Smith, J. Josephson, C. Evans, P. Straum, and J.Noga. Design For a Red-Cell Antibody Identification Expert. Proc. Second International Conf. on Medical Computer Science and Computational Medicine, IEEE Computer Society, Glouster, Ohio, 1983.

13. J. Josephson, B. Chandrasekaran, and J. W. Smith. The Overview Function in Diagnostic Problem Solving. Technical Paper, AI Group, Dept. of Computer and Information Science, The Ohio State University, 1984.

FIGURE 1

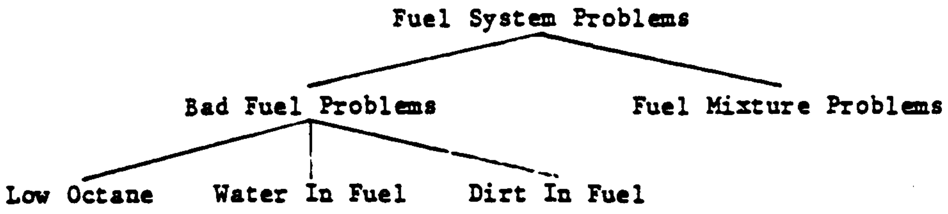


FIGURE 2

```
(Specialist BadFuel
    (declare (superspecialist FuelSystem)
             (subspecialists LowOctane WaterInFuel DirtInFuel))
    (kgs ...)
    (messages ...))
```

FIGURE 3

```
(Establish (if (GE relevant 0)
               then (SetConfidence self summary)
             else (SetConfidence self relevant)))
```

FIGURE 4

```
(Refine (for specialist in subspecialists
           do (Call specialist with Establish)
              (if (+? specialist)
                  then (Call specialist with Refine))))
```

FIGURE 5

```
(relevant Table
     (match (AskYNU? "Is the car slow to respond")
            (AskYNU? "Does the car start hard")
            (And (AskYNU? "Do you hear knocking or pinging sounds")
                 (AskYNU? "Does the problem occur while accelerating"))
       with (if T ? ?
                 then -3
               elseif ? T ?
                 then -3
               elseif ? ? T
                 then 3
               else 1)))
```

FIGURE 6

```
(summary Table
    (match relevant gas
      with (if 3 (GE 0)
               then 3
             elseif 1 (GE 0)
               then 2
             elseif ? (LT 0)
               then -3)))
```

APPENDIX 3

To appear in Proceedings of the 1984 IEEE Expert Systems Principles Workshop, Denver, Colorado.

Assembling the Best Explanation

John R. Josephson, B. Chandrasekaran, Jack W. Smith, Jr.

Artificial Intelligence Group and Department of Pathology
The Ohio State University

O.S.U. AI GROUP TECHNICAL NOTE
26 June 1984

Address correspondence to:
John R. Josephson
The Ohio State University
Dept. of Computer and Information Science
228 C&AE Bldg.
2036 Neil Ave. Mall
Columbus, Ohio 43210

Table of Contents

Assembling the Best Explanation

ABSTRACT

Going from data describing a situation to an explanatory hypothesis that best accounts for the data is a commonly occurring knowledge-based reasoning problem. Sometimes the need is to assemble interacting hypothesis parts into a unified hypothesis. In a medical diagnosis, for example, there might be several diseases present, and they might be related causally. Disease hypotheses sometimes overlap in what they can explain.

In this paper we describe a general mechanism for accomplishing the unification of sub-hypotheses with possibly overlapping domains of explanation. This mechanism makes use of plausibility information concerning the sub-hypotheses, along with information about what a sub-hypothesis can explain in the particular situation, to build towards a complete explanation. The novel capability arises of confirming a sub-hypothesis on the basis of its ability to explain some feature for which there is no other plausible explanation.

Hypothesis interactions are considered to be of two general types, each with its own kind of significance for the problem-solving:

- Explanatory interactions, i.e. due to overlapping in what they can account for.
- Substantive interactions of mutual support and incompatibility, e.g. resulting from causal, logical, or definitional relations[1].

The mechanism we describe can accommodate additive cooperation in accounting for the features of the situation. While this is not yet general enough to handle all types of explanatory interaction, it is nevertheless more general than the set-covering model which considers that an hypothesis either fully accounts for a feature or it does not. [9] The

[1]Two disease hypotheses might offer to explain the same findings without being especially compatible or incompatible causally, logically, or definitionally. On the other hand, hypotheses might be mutually exclusive (e.g. because they represent distinct sub-types of the same disease). These two senses in which hypotheses may be said to be "alternatives" need to be distinguished, so that the problem solving can be organized appropriately.

mechanism described here also accommodates substantive hypothesis interactions of mutual compatability and incompatibility, and interactions of the sort where one hypothesis, if it is accepted, suggests some other hypothesis. Prospects seem good for extending the mechanism to accommodate other forms of interaction too.

An earlier and more primitive version of this mechanism has been used successfully as the basis for an expert system, RED, designed to solve real-world problems of red-cell antibody identification [10]. These are problems which arise in the hospital blood bank, and are solved by specially trained human experts.

AN ARCHITECTURE FOR ABDUCTION USING OVERVIEW

> 'There is no great mystery in this matter', he said, taking the cup of tea which I had poured out for him; 'the facts appear to admit of only one interpretation.' [3]
>
> —— Sherlock Holmes

The philosopher Charles Sanders Peirce has described reasoning that goes from data to a hypothesis which explains the data as a form of inference distinct from both deduction and induction [6]. He calls this third form abduction, and we adopt this term here.

In some problem situations abduction can be accomplished by a relatively simple classification, or hypothesize-and-match mechanism. If the number of potentially applicable hypotheses is small, and if only one can be correct, then each hypothesis can be matched against the data, with the quality of the matchings determining the winning hypothesis. But if more than one can be correct, and if the number of potentially applicable hypotheses is at all large, then the combinatorics of the situation will not permit us to have one pre-established pattern for each possible conclusion.

The alternative seems to be to actively construct the abductive conclusion as a combination of sub-hypotheses which are either abductive conclusions themselves, or are the results from some classification mechanism working from pre-established patterns. Since the breakdown into sub-hypotheses cannot go on _ad infinitum_, in the end there would seem to be no escape from the need at some level for primitive pre-established categories to assemble into an abductive conclusion. Thus we propose an architecture for abduction which consists of two main cooperating modules:

- a module for selecting sub-hypotheses appropriate to the case at hand, and
- a module, which we call Overview, for assembling these sub-hypotheses into the overall best available conclusion for the case.

Overview and the other module communicate through a shared language of the plausibility of sub-hypotheses, and of the findings that are to be explained.

SELECTING APPROPRIATE SUB-HYPOTHESES

A primary function of the sub-hypothesis selection mechanism is to rule out of consideration those of its stored sub-hypotheses as can be determined to be either incompatible with the evidence, or irrelevant to explaining the data of the case. Each sub-hypothesis that cannot be ruled out must be matched against the data to produce a description of what parts of the data it can explain (or contribute to explaining), and how plausible it is under the circumstances.

This plausibility estimate is not an estimate of its probability or certainty of being true in the case. An estimate of this sort would need to take account of interactions between the sub-hypotheses (from a global perspective), because it needs to be based upon such considerations as whether there are available alternative ways of explaining things, and whether this particular sub-hypothesis is contrary to another one under consideration. In our design we assign the global perspective to Overview. The plausibility estimate that Overview needs from the selection mechanism is that a sub-hypothesis could be true, or that it is worth pursuing, based upon the quality of the match (local perspective) between the sub-hypothesis and that part of the data which is specifically relevant to this plausibility estimate. Thus, in contradistinction to the Internist system, in our system no part of the confidence initially ascribed to a hypothesis is based upon what it fails to explain [7].

Another significant difference from Internist is that, on our approach, a clear distinction is made between matching a sub-hypothesis with the data in order to to confirm or rule out, and matching to see what can be explained. Internist's frequency weights of manifestations seem to be doing the work of providing some of the confirmatory evidence, and also of providing information about what is already explained by the hypothesis when it comes time to focus on the remaining unexplained finding [8].

The primary purpose of this paper to concentrate on the Overview module, so a careful comparison will not be undertaken here of the various approaches to hypothesize-and-match mechanisms. The reader is referred to [11], [1], and [5] for some of the alternatives. But it should be pointed out that the establish-refine problem-solving regime of the MDX family of expert systems [2] stands out as especially appropriate to the needs of this architecture.

Establish-refine would be appropriate whenever the sub-hypotheses are naturally organized into hierarchies of more general and more specific hypotheses. When this is the case, advantage can be taken of it in two ways:

- in pruning the search for appropriate sub-hypotheses, where ruling out a general hypothesis should rule out any of its more specific refinements, and
- where ruling out cannot be done, Overview can use information about what still needs to be explained to help make decisions about which sub-hypotheses to pursue in more detail.

Reciprocally, the need for an Overview for the MDX family, similar to the one here proposed, was envisioned in [4].

THE INFORMATION PROCESSING TASK OF OVERVIEW

Abstractly, Overview's job can be thought of as conducting a search through a space of global-hypotheses (i.e. assemblies of sub-hypotheses), where a goal node is a best explanation for the data. A typical link is the addition of a sub-hypothesis to the embly with the object of making the explanation more complete.

Overview is presented with information of various kinds:

1. the data to be explained,

2. a set of hypotheses that plausibly apply to the case,

3. information, particularized to the case, about what each hypothesis can explain,

4. a plausibility rating for each hypothesis, and

5. information about substantive interactions between hypotheses.

Overview's first task is to use this information in order to assemble, if possible, a plete account of the data. The assembly proceeds in such a way that it respects the usibility information, i.e. prefers higher plausibility sub-hypotheses to lower. A ber of different search regimes could be used for the assembly process, but for a eral reasons we have chosen to drive the search by the goal of explaining the most nificant unexplained finding. There is much to be said about the details of this cess, but time will not permit just now.

Once complete explanation is assembled, explanatorially superfluous parts are removed ain, respecting plausibilities) in order to make the assembled hypothesis parsimonious. n it is examined to determine which of its parts are "globally indispensable". An othesis is classified this way if it provides an explanation for some feature of the uation that cannot be plausibly explained by any other hypothesis or combination of otheses. Such an indispensable hypotheses, if it is reasonably plausible to start h, is a prime candidate for being considered as "abductively confirmed", needing only additional support that the case is overall tidy enough, and the evidence broad ugh, for the drawing of confident conclusions.

Thus the standard for abductive confirmation is that the hypothesis be:

- part of a complete, parsimonious, and tidy account of the case,
- intrinsically reasonably plausible (i.e. locally well-matched to the case), and

- indispensable (i.e. there is some feature for which there is no other plausible explanation).

THE DOMAIN OF RED

Red blood cells have on their surfaces certain substances, antigens, which can provoke the immune system of a transfusion recipient to produce antibodies. When it is anticipated that someone may be needing blood products containing red cells, it is especially important to know about antibodies, already circulating in the patient's blood, which are capable of immediately attacking the new red cells. Such antibodies most probably would be present because of previous transfusions, but they can occur naturally. When the patient's antibodies are known, a unit of blood can be chosen to administer which has red cells with antigens that will not provoke an immediate response.

In order to screen for the presence of circulating antibodies, a small amount of the patient's blood serum is mixed with certain screening cells, chosen especially to have on their surface a full range of antigens, to see if any reaction is provoked. If a reaction is provoked, then more tests are performed to determine precisely what circulating antibodies are present. The first step is to do a "panel" which consists of mixing ten or so "cells" (i.e. specimens of identical cells) with the patient's serum in each of five or so different testing conditions. Thus approximately 50 individual tests are involved in a panel. The test cells in the panel, which are usually provided by a manufacturer of medical laboratory materials, each have certain known antigens on their surfaces. The presence or absence of approximately 30 significant antigens is known about each test cell. These known antigens may be expressed with varying strengths depending on the genetic makeup of the cell, and most of this information about the strength of expression can be inferred from the other antigens present on the cell. Any reactions that occur when the panel is done are graded by the technologist performing the panel as to strength and type of reaction. Thus the information from a panel consists of 50 or so reactions (counting non-reaction as a kind of reaction), each one graded into one of 7 or so strengths or types, on 10 or so red cells, under 5 or so different test conditions, each cell having some subset of 30 or so antigens whose strength of expression on the cell might be one of 2 or 3 different grades.

The task of RED is to digest this information, and produce the best possible conclusion on the available evidence of what antigens on the test cells the patient's antibodies are recognizing and attacking. If the results are inconclusive, and there remain antibodies whose presence has been neither ruled out nor confirmed, then a pragmatic medical decision is called for as to whether a unit of blood can now be chosen, or whether additional tests should be performed to resolve the remaining ambiguities.

THE ARCHITECTURE OF RED

RED consists of three main modules:

- An intelligent data base of patient information, panel results, and information about the antigenic makeup of the cells that are used in the panel. (This will not be discussed in this paper except to say that the inferencing concerning the strength of expression of a particular antigen on a cell is done by this module.)
- A community of antibody specialists, one for each antibody and each antibody sub-type, which provides the needed sub-hypothesis selection mechanism (each antibody sub-type represents a distinct sub-hypothesis), and
- The Overview module that unites the viable sub-hypotheses into an overall judgment concerning the case.

THE PERFORMANCE OF RED

The following is an edited transcript of the output of the first version of RED for a case where the first panel performed was not conclusive, and further testing had to be done to settle the remaining ambiguities of the case. This output represents the state at the end of the first inconclusive panel. The correct answer after further testing turned out to be: ANTI-FY-A, ANTI-K, and ANTI-D, i.e. antibodies to the antigens FY-A, K, and D. As can be seen, RED anticipated the correct answer to the antibody problem, although it was unable to confirm it.

The set of antibodies with specificities to (FY-A K E D C) appears sufficient to explain all of the test results.

Antibody towards FY-A is not needed for explaining the test results.

Antibody towards K is needed for explaining
((PANCO-8 ENZYME-37 1)
(PANCO-8 ENZYME-IS 1) (PANCO-8 ALBUMIN-COOMBS 2)
(PANCO-8 ALBUMIN-37 1))

[COMMENT: These are test results encoded as (cell, test conditions, reaction strength) triples.]

Antibody towards E is not needed for explaining the test results.

Antibody towards D is needed for explaining
((PANCO-9 ENZYME-37 1) (PANCO-9 ALBUMIN-37 1)
(PANCO-9 ALBUMIN-IS 1))

Antibody towards C is not needed for explaining the test results.

ANTIBODY STATUS REPORT:

The following antibodies have been ruled out: (ANTI-LU-B ANTI-SMALL-S ANTI-S ANTI-M ANTI-P1 ANTI-LE-A ANTI-JK-B ANTI-JK-A ANTI-FY-B ANTI-JS-B ANTI-KP-B ANTI-S ANTI-SMALL-K ANTI-V ANTI-SMALL-F ANTI-SMALL-E ANTI-SMALL-C)

The following antibodies have been confirmed to be present:
(ANTI-K ANTI-D).

[COMMENT: These were actually present, as was confirmed by later tests. This is really a weak confirmation, since only a weak form of the "Rule of Three"[2] was passed.]

PLEASE NOTE:
The presence of these antibodies is INSUFFICIENT to explain all of the test results.

The following antibody is unresolved but is likely to be present: (ANTI-FY-A).

[COMMENT: This is the third antibody that turned out to be present.]

---

[2] a condition requiring evidential breadth: that there must be at least three cells with the antigen that reacted, and three cells without the antigen that did not react

PLEASE NOTE:
The antibodies considered to be unresolved but likely to be present, and those considered to be confirmed as present, together are SUFFICIENT to explain all of the test results.

The following antibodies are unresolved and are likely to be absent: (ANTI-LU- A ANTI-LE-B ANTI-CW ANTI-E).

[COMMENT: They turned out to be absent.]

The following antibodies are unresolved: (ANTI-N ANTI-C).

[COMMENT: Really absent.]

Type the name of an antibody if you would like an explanation of why it was classified the way it was.

>>>>ANTI-FY-A

was classified as Unresolved But Likely Present because, it was rated at high plausibility by its antibody specialist, and its presence is a good way to explain some reactions.

DISCUSSION

In this paper we have given an account of some aspects of the overview process that is useful in assembling, unifying and synthesizing a best-account abductive hypothesis from a number of plausible hypothesis fragments. We have described only a few of the possible hypothesis interactions that can play a role in this process. There are a number of other kinds of interaction that are important to take into account. In particular we have said nothing yet about causal interactions, or the existence of functional relations between hypotheses, or how statistical correlations may be accommodated. The assembly process will need to be enriched to take all these possibilities into account.

Overview also has the capability to help the sub-hypothesis generation process in a mutually reinforcing manner. E.G., the establish-refine process may have "suspended" [4] a number of general hypotheses due to lack of confirmatory or disconfirmatory evidence of a strong enough nature. The Overview process has the capability to use what remains to be explained as a guide to selectively call on the refinements of some of the suspended hypotheses to try to establish themselves.

REFERENCES

[1] Aikins, J.
Prototypical Knowledge for Expert Systems.
Artificial Intelligence 20:163-210, 1983.

[2] Chandrasekaran, B. / Mittal, S.
Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems.
In M. Yovits (editor), Advances in Computers, , pages 217-293. Academic Press, 1983.

[3] Doyle, Sir Arthur Conan.
The Sign of the Four.
Clarkson N. Potter, Inc., 1960, .

[4] Gomez, F. / Chandrasekaran, B.
Knowledge Organization and Distribution for Medical Diagnosis.
IEEE Transactions on Systems, Man, and Cybernetics SMC-11(1):34-42, January, 1981.

[5] Pauker SG, Gorry GA, Kassirer JP and Schwartz WB.
Towards the simulation of clinical cognition: Taking a present illness by computer.
Amer J Med 60:981-996, 1975.

[6] Peirce, C.S.
Abduction and Induction.
Dover, 1955, pages 151chapter 11.

[7] Pople H, Myers J, and Miller R.
DIALOG: A model of diagnostic logic for internal medicine.
In Proc 4th Int Joint Conf Artificial Intell, pages 848-855. International Joint Conferences on Artificial Intelligence, Tbilisi, 1975.

[8] Pople, H.
The Formation of Composite Hypotheses in Diagnotice Problem Solving.
In Proceedings of IJCAI, 1977, pages 1030-1037. IJCAI, 1977.

[9] Reggia,J.A.; Nau,D.S.; and Wang,P.
Diagnostic Expert Systems Based on a Set Covering Model.
Internat. Jour. Man-Machine Studies :?, Oct, 1982.

[10] J. W. Smith, J. Josephson, C. Evans, P. Straum, and J.Noga.
Design For a Red-Cell Antibody Identification Expert.
In Proc. Second International Conf. on Medical Computer Science and Computational Medicine. 1983.

[11] Szolovits P and Pauker SG.
Categorical and probabilistic reasoning in medical diagnosis.
Artificial Intelligence 11:115-144, 1978.

END

FILMED

10-85

DTIC